Approximating Parameterized Convex Optimization Problems *

Joachim Giesen
Friedrich-Schiller-Universität Jena, Germany

Martin Jaggi
ETH Zürich, Switzerland

Sören Laue
Friedrich-Schiller-Universität Jena, Germany

August 10, 2010

Abstract

We consider parameterized convex optimization problems over the unit simplex, that depend on one parameter. We provide a simple and efficient scheme for maintaining an ε-approximate solution (and a corresponding ε-coreset) along the entire parameter path. We prove correctness and optimality of the method. Practically relevant instances of the abstract parameterized optimization problem are for example regularization paths of support vector machines, multiple kernel learning, and minimum enclosing balls of moving points.

1 Introduction

We study convex optimization problems over the unit simplex that are parameterized by a single parameter. We are interested in optimal solutions of the optimization problem for all parameter values, i.e., the whole solution path in the parameter. Since the complexity of the exact solution path might be exponential in the size of the input [7], we consider approximate solutions with an approximation guarantee for all parameter values, i.e., approximate solutions along the whole path. We provide a general framework for computing approximate solution paths that has the following properties:

*A preliminary version of this article appeared in Proceedings of the 18th European Symposium on Algorithms, 2010.The research of J. Giesen and S. Laue is supported by the DFG (grant GI-711/3-1). The research of M. Jaggi is supported by a Google Research Award and by the Swiss National Science Foundation (SNF grant 20PA21-121957).

(1) *Generality.* Apart from being specified over the unit simplex, we hardly make any assumptions on the optimization problem under consideration. Hence, the framework can be applied in many different situations.

(2) *Simplicity.* The basic idea behind the framework is a very simple continuity argument.

(3) *Practicality.* We show that our framework works well for real world problems.

(4) *Efficiency.* Although the framework is very simple it still gives improved theoretical bounds for known problems.

(5) *Optimality.* We show that it is the best possible one can do up to a constant factor.

Let us explain the different aspects in more detail.

Generality: We build on the general primal-dual approximation criterion that has been introduced by Clarkson in his coreset framework [5] for convex optimization problems over the unit simplex. Among the many problems that fit into Clarkson's framework are for example the smallest enclosing ball problem, polytope distance problems, binary classification support vector machines, support vector regression, multiple kernel learning, AdaBoost, or even mean-variance analysis in portfolio selection [12]. For almost all of these problems, parameterized versions are known and important to consider, e.g. the smallest enclosing ball problem for points that move with time, or soft margin support vector machines which trade-off a regularization term and a loss term in the objective function of the optimization problem.

Simplicity: The basic algorithmic idea behind our framework is computing at some parameter value an approximate solution whose approximation guarantee holds for some sub-interval of the problem path. This solution is then updated at the boundary of the sub-interval to a better approximation that remains a good approximation for a consecutive sub-interval. For computing the initial approximation and the updates from previous approximations, any arbitrary (possibly problem specific) algorithm can be used, that ideally can be started from the previous solution (warm start). We provide a simple lemma that allows to bound the number of necessary parameter sub-intervals for a prescribed approximation quality. For interesting problems, the lemma also implies the existence of small coresets that are valid for the entire parameter path.

Practicality: Our work is motivated by several problems from machine learning and computational geometry that fit into the described framework, in particular, support vector machines and related classification methods, multiple kernel learning [3], and the smallest enclosing ball problem [4]. We have implemented the proposed algorithms and applied them to choose the optimal regularization parameter for a support vector machine, and to find the best combination of two kernels which is a special case of the multiple kernel learning problem.

Efficiency: Our framework gives a path complexity of $O\left(\frac{1}{\varepsilon}\right)$, meaning that an ε-approximate solution needs to be updated only $O\left(\frac{1}{\varepsilon}\right)$ times along the whole path. This positively contrasts the complexity of exact solution paths.

Optimality: We provide lower bounds that show that one cannot do better, i.e., there exist examples where one needs essentially at least one fourth as many sub-intervals as predicted by our method.

Related Work Many of the aforementioned problems have been recently studied intensively, especially machine learning methods such as computing exact solution paths in the context of support vector machines and related problems [11, 16, 18, 8]. But exact algorithms can be fairly slow compared to approximate methods as they need to invert large matrices. To make things even worse, the complexity of exact solution paths can be very large, e.g., it can grow exponentially in the input size as it has been shown for support vector machines with ℓ_1-loss [7]. Hence, approximation algorithms have become popular also for the case of solution paths lately, see e.g. [6]. However, to the best of our knowledge, so far no approximation quality guarantees along the path could be given for any of these existing algorithms.

2 Clarkson's Framework

In [5] Clarkson considers convex optimization problems of the form

$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t.} & x \in S_n \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and continuously differentiable, and S_n is the unit simplex, i.e., S_n is the convex hull of the standard basis vectors of $\mathbb{R}^n$. We additionally assume that the function f is non-negative on S_n. A point $x \in \mathbb{R}^n$ is called a feasible solution, if $x \in S_n$.

The Lagrangian dual of Problem 1 (sometimes also called Wolfe dual) is given by the unconstrained problem

$$\max_x \omega(x), \text{ where } \omega(x) := f(x) + \min_i (\nabla f(x))_i - x^T \nabla f(x).$$

In this framework Clarkson studies approximating the optimal solution. His measure of approximation quality is (up to a multiplicative positive constant) the primal-dual gap

$$g(x) := f(x) - \omega(x) = x^T \nabla f(x) - \min_i (\nabla f(x))_i.$$

Note that convexity of f implies the weak duality condition $f(\hat{x}) \geq \omega(x)$, for the optimal solution $\hat{x} \in S_n$ of the primal problem and any feasible solution x, which in turn implies non-negativity of the primal-dual gap, i.e., $g(x) \geq 0$ for all feasible x, see [5].

Definition 1 *A feasible solution x is an* ε-approximation *to Problem 1 if*

$$g(x) \leq \varepsilon f(x).$$

A subset $C \subseteq [n]$ is called an ε-coreset*, if there exists an ε-approximation x to Problem 1 with $x_i = 0$, $\forall i \in [n] \setminus C$.*

Sometimes in the literature, a multiplicative ε-approximation is defined more restrictively as $g(x) \leq \varepsilon f(\hat{x})$, relative to the optimal value $f(\hat{x})$ of the primal optimization problem. Note that this can directly be obtained from our slightly weaker definition by setting ε in the definition of an ε-approximation to $\varepsilon' := \frac{\varepsilon}{1+\varepsilon}$, because $g(x) \leq \frac{\varepsilon}{1+\varepsilon} f(x) \Leftrightarrow (1+\varepsilon)(f(x) - \omega(x)) \leq \varepsilon f(x) \Leftrightarrow g(x) \leq \varepsilon\omega(x) \leq \varepsilon f(\hat{x})$.

The case of *maximizing* a concave, continuously differentiable, non-negative function f over the unit simplex S_n can be treated analogously. The Lagrangian dual problem is given as

$$\min_x \omega(x), \text{ where } \omega(x) := f(x) + \max_i (\nabla f(x))_i - x^T \nabla f(x),$$

and the duality gap is $g(x) := \omega(x) - f(x) = \max_i (\nabla f(x))_i - x^T \nabla f(x)$. Again, $x \in S_n$ is an ε-approximation if $g(x) \leq \varepsilon f(x)$ (which immediately implies $g(x) \leq \varepsilon f(\hat{x})$ for the optimal solution $\hat{x}$ of the primal maximization problem).

Clarkson [5] showed that ε-coresets of size $\left\lceil \frac{2C_f}{\varepsilon} \right\rceil$ do always exist, and that the sparse greedy algorithm [5, Algorithm 1.1] obtains an ε-approximation after at most $2\left\lceil \frac{4C_f}{\varepsilon} \right\rceil$ many steps. Here C_f is an absolute constant describing the "non-linearity" or "curvature" of the function f.

3 Optimizing Parameterized Functions

We extend Clarkson's framework and consider parameterized families of functions $f_t(x) = f(x;t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ that are convex and continuously differentiable in x and parameterized by $t \in \mathbb{R}$, i.e., we consider the following families of minimization problems

$$\begin{array}{ll} \min_x & f_t(x) \\ \text{s.t.} & x \in S_n \end{array} \tag{2}$$

Again, we assume $f_t(x) \geq 0$ for all $x \in S_n$ and $t \in \mathbb{R}$. We are interested in ε-approximations for all parameter values of $t \in \mathbb{R}$.

The following simple lemma is at the core of our discussion and characterizes how we can change the parameter t such that a given $\frac{\varepsilon}{\gamma}$-approximate solution x (for $\gamma > 1$) at t stays an ε-approximate solution.

Lemma 2 *Let $x \in S_n$ be an $\frac{\varepsilon}{\gamma}$-approximation to Problem 2 for some fixed parameter value t, and for some $\gamma > 1$. Then for all $t' \in \mathbb{R}$ that satisfy*

$$\begin{aligned} x^T \nabla(f_{t'}(x) - f_t(x)) - (\nabla(f_{t'}(x) - f_t(x)))_i - \varepsilon(f_{t'}(x) - f_t(x)) \\ \leq \varepsilon\left(1 - \tfrac{1}{\gamma}\right) f_t(x), \qquad \forall i \in [n], \end{aligned} \tag{3}$$

the solution x is still an ε-approximation to Problem 2 at the changed parameter value t'.

Proof: We have to show that $g_{t'}(x) \leq \varepsilon f_{t'}(x)$, or in other words that

$$x^T \nabla f_{t'}(x) - (\nabla f_{t'}(x))_i \leq \varepsilon f_{t'}(x)$$

holds for all components i. We add to the Inequalities 3 for all components i the inequalities stating that x is an $\frac{\varepsilon}{\gamma}$-approximate solution at value t, i.e.

$$x^T \nabla f_t(x) - (\nabla f_t(x))_i \leq \frac{\varepsilon}{\gamma} f_t(x).$$

This gives for all $i \in [n]$

$$x^T \nabla f_{t'}(x) - (\nabla f_{t'}(x))_i - \varepsilon(f_{t'}(x) - f_t(x)) \leq \varepsilon f_t(x),$$

which simplifies to the claimed bound $x^T \nabla f_{t'}(x) - (\nabla f_{t'}(x))_i \leq \varepsilon f_{t'}(x)$ on the duality gap at t'. □

The analogue of Lemma 2 for *maximizing* a concave function over the unit simplex is the following lemma whose proof follows along the same lines:

Lemma 3 *Let $x \in S_n$ be an $\frac{\varepsilon}{\gamma}$-approximation to the maximization problem $\max_{x \in S_n} f_t(x)$ at parameter value t, for some $\gamma > 1$. Here $f_t(x)$ is a parameterized family of concave, continuously differentiable functions in x that are non-negative on S_n. Then for all $t' \in \mathbb{R}$ that satisfy*

$$\begin{aligned} (\nabla(f_{t'}(x) - f_t(x)))_i - x^T \nabla(f_{t'}(x) - f_t(x)) - \varepsilon(f_{t'}(x) - f_t(x)) \\ \leq \varepsilon \left(1 - \tfrac{1}{\gamma}\right) f_t(x), \qquad \forall i \in [n], \end{aligned} \tag{4}$$

the solution x is still an ε-approximation at the changed parameter value t'.

Definition 4 *The* ε-approximation path complexity *of Problem 2 is defined as the minimum number of sub-intervals over all possible partitions of the parameter range $\mathbb{R}$, such that for each individual sub-interval there is a single solution of Problem 2 which is an ε-approximation for that entire sub-interval.*

Lemma 2 and 3 imply upper bounds on the path complexity. Next, we will show that these upper bounds are tight up to a multiplicative factor of $4 + 2\varepsilon$.

3.1 Lower Bound

To see that the approximate path complexity bounds we get from Lemma 3 are optimal consider the following parameterized optimization problem:

$$\begin{array}{ll} \max_x & f_t(x) := x^T f(t) \\ \text{s.t.} & x \in S_n \end{array} \tag{5}$$

where $f(t) = (f_0(t), \ldots, f_{n-1}(t))$ is a vector of functions and $f_i(t)$ is defined as follows

$$f_i(t) = \begin{cases} 0, & \text{for } t < i\varepsilon' \\ t - i\varepsilon', & \text{for } i\varepsilon' \leq t < 1 + i\varepsilon' \\ -t + 2 + i\varepsilon', & \text{for } 1 + i\varepsilon' \leq t \leq 2 + i\varepsilon' \\ 0, & \text{for } 2 + i\varepsilon' < t \end{cases}$$

for some arbitrary fixed $\varepsilon' > 0$ and $n > 1/\varepsilon'$. See Figure 1 for an illustration of the function $f_i(t)$.

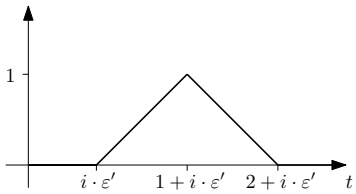

Figure 1: Function $f_i(t)$.

Each of the $f_i(t)$ attains its maximum 1 at $t = 1 + i\varepsilon'$. Since $f_t(x)$ is linear in x it is also concave in x for every fixed t. Hence, it is an instance of Problem 2. Let us now consider the interval $t \in [1, 2]$. In this interval consider the points $t_i := 1 + i\varepsilon'$, for $i = 0, \ldots, \lfloor 1/\varepsilon' \rfloor$. At each of these points it holds that $f_i(t_i) = 1$ and all the other $f_j(t_i) \leq 1 - \varepsilon'$ when $j \neq i$. Hence, the value of the optimal solution to Problem 5 at parameter value t_i is 1, and it is attained at $x = e_i$, where e_i is the i-th standard basis vector. Furthermore, for all other $x \in S_n$ that have an entry at the coordinate position i that is at most $1/2$ it holds that $f_{t_i}(x) \leq 1 - \varepsilon'/2$.

Hence, in order to have an ε-approximation for $\varepsilon < \varepsilon'/2$ the approximate solution x needs to have an entry of more than $1/2$ at the i-th coordinate position. Since all entries of x sum up to 1, all the other entries are strictly less than $1/2$ and hence this solution cannot be an ε-approximation for any other parameter value $t = t_j$ with $j \neq i$. Thus, for all values of $t \in [1, 2]$ one needs at least $1/\varepsilon'$ different solutions for any $\varepsilon < \varepsilon'/2$.

Choosing ε' arbitrarily close to 2ε this implies that one needs at least $\frac{1}{2\varepsilon} - 1$ different solutions to cover the whole path for $t \in [1, 2]$.

Lemma 3 gives an upper bound of $\frac{2+\varepsilon}{\varepsilon} \frac{\gamma}{\gamma - 1} = \left(\frac{2}{\varepsilon} + 1\right) \frac{\gamma}{\gamma - 1}$ different solutions, since $\nabla f_t(x) = f(t) = (f_i(t))_{i \in [n]}$ and $\left|\frac{\partial f_i}{\partial t}\right| \leq 1$, $\left(\nabla(f_{t'}(x) - f_t(x))\right)_i \leq |t' - t|$. Hence, this is optimal up to a factor of $4 + 2\varepsilon$. Indeed, also the dependence on the problem specific constants in Lemma 3 is tight: 'contracting' the functions $f_i(t)$ along the t-direction increases the Lipschitz constant of $(\nabla f_t(x))_i$, which is an upper bound on the problem specific constants in Lemma 3.

3.2 The Weighted Sum of Two Convex Functions

We are particularly interested in a special case of Problem 2. For any two convex, continuously differentiable functions $f^{(1)}, f^{(2)} : \mathbb{R}^n \to \mathbb{R}$ that are non-negative on S_n, we consider the weighted sum $f_t(x) := f^{(1)}(x) + tf^{(2)}(x)$ for a real parameter $t \geq 0$. The parameterized optimization Problem 2 in this case becomes

$$\begin{array}{ll} \min_x & f^{(1)}(x) + tf^{(2)}(x) \\ \text{s.t.} & x \in S_n \end{array} \tag{6}$$

For this optimization problem we have the following corollary of Lemma 2:

Corollary 5 *Let $x \in S_n$ be an $\frac{\varepsilon}{\gamma}$-approximate solution to Problem 6 for some fixed parameter value $t \geq 0$, and for some $\gamma > 1$. Then for all $t' \geq 0$ that satisfy*

$$(t' - t)\left(x^T \nabla f^{(2)}(x) - (\nabla f^{(2)}(x))_i - \varepsilon f^{(2)}(x)\right) \leq \varepsilon \left(1 - \frac{1}{\gamma}\right) f_t(x), \quad \forall i \in [n] \tag{7}$$

solution x is an ε-approximate solution to Problem 6 at the parameter value t'.

Proof: Follows directly from Lemma 2, and $f_{t'}(x) - f_t(x) = (t' - t)f^{(2)}(x)$. □

This allows us to determine the entire interval of admissible parameter values t' such that an $\frac{\varepsilon}{\gamma}$-approximate solution at t is still an ε-approximate solution at t'.

Corollary 6 *Let x be an $\frac{\varepsilon}{\gamma}$-approximate solution to the Problem 6 for some fixed parameter value $t \geq 0$, for some $\gamma > 1$, and let*

$$\begin{aligned} u &:= x^T \nabla f^{(2)}(x) - \min_i \left(\nabla f^{(2)}(x)\right)_i - \varepsilon f^{(2)}(x) \\ l &:= x^T \nabla f^{(2)}(x) - \max_i \left(\nabla f^{(2)}(x)\right)_i - \varepsilon f^{(2)}(x), \end{aligned}$$

then $l \leq u$ and x remains an ε-approximate solution for all $0 \leq t' = t + \delta$ for the following values of δ:

(i) If $l < 0$ and $0 < u$, then the respective admissible values for δ are

$$\varepsilon \left(1 - \frac{1}{\gamma}\right) \frac{f_t(x)}{l} \leq \delta \leq \varepsilon \left(1 - \frac{1}{\gamma}\right) \frac{f_t(x)}{u}$$

(ii) If $u \leq 0$, then δ (and thus t') can become arbitrarily large.

(iii) If $l \geq 0$, then δ can become as small as $-t$, and thus t' can become 0.

Note that the ε-approximation path complexity for Problem 6 for a given value of $\gamma > 1$ can be upper bounded by the minimum number of points $t_j \geq 0$ such that the admissible intervals of $\frac{\varepsilon}{\gamma}$-approximate solutions x_j at t_j cover the

whole parameter interval $[0, \infty)$.

Corollary 6 immediately suggests two variants of an algorithmic framework (forward- and backward version) maintaining ε-approximate solutions over the entire parameter interval, or in other words, tracking a guaranteed ε-approximate solution path. Note that as the internal optimizer, any arbitrary approximation algorithm can be used here, as long as it provides an approximation guarantee on the relative primal-dual gap. For example the standard Frank-Wolfe algorithm [5, Algorithm 1.1] is particularly suitable as its resulting coreset solutions are also sparse. The forward variant is depicted in Algorithm 1 and the backward variant in Algorithm 2.

Algorithm 1 ApproximationPath—ForwardVersion $(\varepsilon, \gamma, t_{\min}, t_{\max})$

1 compute an $\frac{\varepsilon}{\gamma}$-approximation x for $f_t(x)$ at $t := t_{\min}$ using a standard optimizer.
2 **do**
3 $\quad u := x^T \nabla f^{(2)}(x) - \min_i \left(\nabla f^{(2)}(x)\right)_i - \varepsilon f^{(2)}(x)$
4 $\quad$ **if** $u > 0$ **then**
5 $\quad\quad \delta := \varepsilon \left(1 - \frac{1}{\gamma}\right) \frac{f_t(x)}{u} > 0$
6 $\quad\quad t := t + \delta$
7 $\quad\quad$ *improve the* (now still ε-approximate) *solution x for $f_t(x)$ to an at least $\frac{\varepsilon}{\gamma}$-approximate solution by applying steps of any standard optimizer.*
8 $\quad$ **else**
9 $\quad\quad t := t_{\max}$
10 **while** $t < t_{\max}$

Algorithm 2 ApproximationPath—BackwardVersion $(\varepsilon, \gamma, t_{\max}, t_{\min})$

1 compute an $\frac{\varepsilon}{\gamma}$-approximation x for $f_t(x)$ at $t := t_{\max}$ using a standard optimizer.
2 **do**
3 $\quad l := x^T \nabla f^{(2)}(x) - \max_i \left(\nabla f^{(2)}(x)\right)_i - \varepsilon f^{(2)}(x)$
4 $\quad$ **if** $l < 0$ **then**
5 $\quad\quad \delta := \varepsilon \left(1 - \frac{1}{\gamma}\right) \frac{f_t(x)}{l} < 0$
6 $\quad\quad t := t + \delta$
7 $\quad\quad$ *improve the* (now still ε-approximate) *solution x for $f_t(x)$ to an at least $\frac{\varepsilon}{\gamma}$-approximate solution by applying steps of any standard optimizer.*
8 $\quad$ **else**
9 $\quad\quad t := t_{\min}$
10 **while** $t > t_{\min}$

4 Applications

Special cases of Problem 6 or the more general Problem 2 have applications in computational geometry and machine learning. In the following we discuss three of these applications in more detail, namely, regularization paths of support vector machines (SVMs), multiple kernel learning, and smallest enclosing balls of linearly moving points. The first two applications for SVMs are special instances of a parameterized polytope distance problem that we discuss at first.

4.1 A Parameterized Polytope Distance Problem

In the setting of Section 3.2 we consider the case $f^{(1)}(x) := x^T K^{(1)} x$ and $f^{(2)}(x) := x^T K^{(2)} x$, for two positive semi-definite matrices $K^{(1)}, K^{(2)} \in \mathbb{R}^{n \times n}$, or formally

$$\begin{array}{ll} \min_x & f^{(1)}(x) + t f^{(2)}(x) = x^T \left(K^{(1)} + t K^{(2)} \right) x \\ \text{s.t.} & x \in S_n \ . \end{array} \tag{8}$$

The geometric interpretation of this problem is as follows: let $A(t) \in \mathbb{R}^{n \times r}$, $r \leq n$, be the unique matrix such $A(t)^T A(t) = K^{(1)} + tK^{(2)}$ (Cholesky decomposition). The solution $\hat{x}$ to Problem 8 is the point in the convex hull of the column vectors of the matrix $A(t)$ that is closest to the origin. Hence, Problem 8 is a parameterized polytope distance problem. For the geometric interpretation of an ε-approximation in this context we refer to [9]. In the following we will consider two geometric parameters for any fixed polytope distance problem:

Definition 7 *For a positive semi-definite matrix $K \in \mathbb{R}^{n \times n}$, we define*

$$\rho_{(K)} := \min_{x \in S_n} x^T K x \qquad \text{and} \qquad R_{(K)} := \max_i K_{ii}$$

or in other words when considering the polytope associated with K, $\rho_{(K)}$ is the minimum (squared) distance to the origin, and $R_{(K)}$ is the largest squared norm of a point in the polytope. We say that the polytope distance problem $\min_{x \in S_n} x^T K x$ is separable *if $\rho_{(K)} > 0$.*

For the parameterized Problem 8, the two quantities u and l that determine the admissible parameter intervals in Corollary 6 and the step size in both approximate path algorithms take the simpler form

$$u = (2-\varepsilon) x^T K^{(2)} x - 2 \min_i (K^{(2)} x)_i \text{ and } l = (2-\varepsilon) x^T K^{(2)} x - 2 \max_i (K^{(2)} x)_i,$$

since $\nabla f^{(2)}(x) = 2K^{(2)} x$. We can now use the following lemma to bound the path complexity for instances of Problem 8.

Lemma 8 *Let $0 < \varepsilon \leq 1$ and $\gamma > 1$. Then for any parameter $t \geq 0$, the length of the interval $[t - \delta, t]$ with $\delta > 0$, on which an $\frac{\varepsilon}{\gamma}$-approximate solution x to Problem 8 at parameter value t remains an ε-approximation, is at least*

$$l_f(\varepsilon, \gamma) := \frac{\varepsilon}{2} \left(1 - \frac{1}{\gamma} \right) \frac{\rho_{(K^{(1)})}}{R_{(K^{(2)})}} = \Omega(\varepsilon) \ . \tag{9}$$

Proof: For $l = (2-\varepsilon)x^T K^{(2)} x - 2\max_i (K^{(2)}x)_i < 0$, we get from Corollary 6 that the length of the left interval at x is of length at least

$$\varepsilon\left(1 - \frac{1}{\gamma}\right)\frac{f_t(x)}{-l}.$$

For any $t \geq 0$, we can lower bound

$$f_t(x) \geq f^{(1)}(x) = x^T K^{(1)} x \geq \min_{x \in S_n} x^T K^{(1)} x = \rho_{(K^{(1)})},$$

and for $\varepsilon \leq 1$ we can upper bound

$$-l = 2\max_i (K^{(2)}x)_i - (2-\varepsilon)x^T K^{(2)} x \leq 2\max_i (K^{(2)}x)_i,$$

because $f^{(2)}(x) \geq 0$. The value $\max_i (K^{(2)}x)_i = \max_i e_i^T K^{(2)} x$ is the inner product between two points in the convex hull of the columns of the square root of the positive semi-definite matrix $K^{(2)}$ (see the discussion at the beginning of this section). Let these two points be $u, v \in \mathbb{R}^n$. Using the Cauchy-Schwarz inequality we get

$$\begin{aligned}\max_i (K^{(2)}x)_i &= u^T v \leq \sqrt{\|u\|^2\|v\|^2} \leq \frac{1}{2}(\|u\|^2 + \|v\|^2) \\ &\leq \max\{\|u\|^2, \|v\|^2\} \leq \max_{x \in S_n} x^T K^{(2)} x,\end{aligned}$$

where the last expression gives the norm of the longest vector with endpoint in the convex hull of the columns of the square root of $K^{(2)}$. However, the largest such norm (in contrast to the smallest norm) is always attained at a vertex of the polytope, or formally $\max_{x \in S_n} x^T K^{(2)} x = \max_i e_i^T K^{(2)} e_i = \max_i K^{(2)}_{ii} = R_{(K^{(2)})}$. Hence, $-l \leq 2R_{(K^{(2)})}$. Combining the lower bound for $f_t(x)$ and the upper bound for $-l$ gives the stated bound on the interval length. □

Now, to upper bound the approximation path complexity we split the domain $[0, \infty]$ into two parts: the interval $[0, 1]$ can be covered by at most $1/l_f(\varepsilon, \gamma)$ admissible left intervals, i.e., by at most $1/l_f(\varepsilon, \gamma)$ many admissible sub-intervals. We reduce the analysis for the interval $t \in [1, \infty]$ to the analysis for $[0, 1]$ by interchanging the roles of $f^{(1)}$ and $f^{(2)}$. For any $t \geq 1$, x is an ε-approximate solution to $\min_{x \in S_n} f_t(x) := f^{(1)}(x) + tf^{(2)}(x)$ if and only if x is an ε-approximate solution to $\min_{x \in S_n} f'_{t'}(x) := t' f^{(1)}(x) + f^{(2)}(x)$ for $t' = \frac{1}{t} \leq 1$, because the definition of an ε-approximation is invariant under scaling the objective function. Note that by allowing $t = \infty$ we just refer to the case $t' = 0$ in the equivalent problem for $f'_{t'}(x)$ with $t' = \frac{1}{t} \in [0, 1]$. Using the lower bounds on the sub-interval lengths $l_f(\varepsilon, \gamma)$ and $l_{f'}(\varepsilon, \gamma)$ (for the problem for $f'_{t'}(x)$ with $t' \in [0, 1]$) on both sub-intervals we get an upper bound of $\left\lceil \frac{1}{l_f(\varepsilon,\gamma)} \right\rceil + \left\lceil \frac{1}{l_{f'}(\varepsilon,\gamma)} \right\rceil$ on the path complexity as is detailed in the following theorem:

Theorem 9 *Given any $0 < \varepsilon \leq 1$ and $\gamma > 1$, and assuming that the distance problems associated to $K^{(1)}$ and $K^{(2)}$ are both separable, we have that the ε-approximation path complexity of Problem 8 is at most*

$$\frac{\gamma}{\gamma - 1} \left(\frac{R_{(K^{(2)})}}{\rho_{(K^{(1)})}} + \frac{R_{(K^{(1)})}}{\rho_{(K^{(2)})}} \right) \frac{2}{\varepsilon} + 2 = O\left(\frac{1}{\varepsilon}\right) \ .$$

This proof of the path complexity immediately implies a bound on the time complexity of our approximation path Algorithm 1. In particular we obtain a linear running time of $O\left(\frac{n}{\varepsilon^2}\right)$ for computing the global solution path when using [5, Algorithm 1.1] as the internal optimizer.

There are interesting applications of this result, because it is known that instances of Problem 8 include for example computing the solution path of a support vector machine – as the regularization parameter changes – and also finding the optimal combination of two kernel matrices in the setting of kernel learning. We will discuss these applications in the following sections.

4.2 The Regularization Path of Support Vector Machines

Support Vector Machines (SVMs) are well established machine learning techniques for classification and related problems. It is known that most of the practically used SVM variants are equivalent to a polytope distance problem, i.e., finding the point in the convex hull of a set of data points that is closest to the origin [9]. In particular the so called one class SVM with ℓ_2-loss term [17, Equation (8)], and the two class ℓ_2-SVM without offset as well as with penalized offset, see [17, Equation (13)] for details, can be formulated as the following polytope distance problem

$$\begin{array}{ll} \min_x & x^T \left(K + \frac{1}{c}\mathbf{1}\right) x \\ \text{s.t.} & x \in S_n \end{array} \tag{10}$$

where the so called *kernel matrix* K is an arbitrary positive semi-definite matrix consisting of the inner products $K_{ij} = \langle \phi(p_i), \phi(p_j) \rangle$ of the data points $p_1, \ldots, p_n \in \mathbb{R}^d$ mapped into some kernel feature space $\phi(\mathbb{R}^d)$. The parameter c $(= 1/t)$ is called the *regularization parameter*, and controls the trade-off between the regularization and the loss term in the objective function. Selecting the right regularization parameter value and by that balancing between low model complexity and overfitting is a very important problem for SVMs and machine learning methods in general and highly influences the prediction accuracy of the method.

Problem 10 is a special case of Problem 8 with $K^{(2)} = \mathbf{1}$, and in this case the quantities u and l (used in Corollary 6 and the approximate path Algorithm 1 and Algorithm 2 now have the even simpler form

$$u = (2 - \varepsilon) x^T x - 2 \min_i x_i \qquad \text{and} \qquad l = (2 - \varepsilon) x^T x - 2 \max_i x_i,$$

and from Lemma 8 we get the following corollary for the complexity of an approximate regularization path, i.e., the approximation path complexity for Problem 10

Corollary 10 *Given $0 < \varepsilon \leq 1$ and $\gamma > 1$, and assuming that the distance problem associated to K is separable, we have that the ε-approximation path complexity of the regularization parameter path for $c \in [c_{\min}, \infty)$ is at most*

$$\frac{\gamma}{\gamma - 1} \frac{R_{(K)} + c_{\min}}{\rho_{(K)} \cdot c_{\min}} \cdot \frac{2}{\varepsilon} + 2 = O\left(\frac{R_{(K)}}{\rho_{(K)} c_{\min} \cdot \varepsilon}\right) = O\left(\frac{1}{\varepsilon \cdot c_{\min}}\right) \ .$$

Proof: As in the proof of Theorem 9, the number of admissible sub-intervals needed to cover the interval of parameter values $t = \frac{1}{c} \in [0, 1]$ can be bounded by

$$\frac{\gamma}{\gamma - 1} \frac{1}{\rho_{(K)}} \frac{2}{\varepsilon} = O\left(\frac{1}{\varepsilon}\right) \quad ,$$

because $R_{(\mathbf{1})} = \max_i \mathbf{1}_{ii} = 1$.

The interval $t \in [1, 1/c_{\min}]$ or equivalently $c \in [c_{\min}, 1]$ (and $f'_c(x) = x^T \mathbf{1} x + c \cdot x^T K x$) can also be analyzed following the proof of Lemma 8. Only, now we bound the function value as follows

$$f'_c(x) = x^T \mathbf{1} x + c x^T K x \geq c x^T K x \geq c_{\min} \min_{x \in S_n} x^T K x = c_{\min} \rho_{(K)}$$

to lower bound the length of an admissible interval. Hence, the number of admissible intervals needed to cover $[c_{\min}, 1]$ is at most

$$\frac{\gamma}{\gamma - 1} \frac{1}{c_{\min}} \frac{R_{(K)}}{\rho_{(K)}} \frac{2}{\varepsilon}.$$

Adding the complexities of both intervals gives the claimed complexity for the regularization path. □

Of course we could also have used Theorem 9 directly, but using $\rho_{(\mathbf{1})} = \frac{1}{n}$ would only give a complexity bound that is proportional to n. However, if we choose to stay above $c_{\min}$, then we can obtain the better bound as described in the above theorem.

Globally Valid Coresets Using the above Theorem 9 for the number $O\left(\frac{1}{\varepsilon \cdot c_{\min}}\right)$ of intervals of constant solutions, and combining this with the size $O\left(\frac{1}{\varepsilon}\right)$ of a coreset at a fixed parameter value, as e.g. provided by [5, Algorithm 1.1], we can just build the union of those individual coresets to get an ε-coreset of size $O\left(\frac{1}{\varepsilon^2 \cdot c_{\min}}\right)$ that is valid over the entire solution path. This means we have upper bounded the overall number of support vectors used in a solution valid over the entire parameter range $c \in [c_{\min}, \infty)$. This is particularly nice as this number is independent of both the number of data points and the dimension of the feature space, and can easily be constructed by our Algorithms 1 and 2.

In Section 5.1 we report experimental results using this algorithmic framework for choosing the best regularization parameter.

4.3 Multiple Kernel Learning

Another immediate application of the parameterized framework in the context of SVMs is "learning" the best combination of two kernels. This is a special case of the multiple kernel learning problem, where the optimal kernel to be used in a SVM is not known a priori, but needs to be selected out of a set of candidates. This set of candidates is often chosen to be the convex hull of a few given "base" kernels, see for example [3]. In our setting with two given kernel matrices $K^{(1)}$,$K^{(2)}$, the kernel learning problem can be written as follows:

$$\begin{array}{ll} \min_x & x^T \left(\lambda K^{(1)} + (1-\lambda)K^{(2)} + \frac{1}{c}\mathbf{1}\right) x \\ \text{s.t.} & x \in S_n \end{array} \tag{11}$$

where $0 \leq \lambda \leq 1$, is the parameter that we want to learn. To simplify the notation, let us define the matrices $K_c^{(1)} := K^{(1)} + \frac{1}{c}$ and $K_c^{(2)} := K^{(2)} + \frac{1}{c}$. By scaling the objective function by $1/\lambda$ (where λ is assumed to be non-zero), Problem 11 can be transformed to a special case of Problem 8, where $t = \frac{1-\lambda}{\lambda}$ (note again that the scaling does not affect our measure of primal-dual approximation error):

$$\begin{array}{ll} \min_x & x^T K_c^{(1)} x + t \cdot x^T K_c^{(2)} x \\ \text{s.t.} & x \in S_n \end{array} \tag{12}$$

This again allows us to apply both approximation path Algorithms 1 and 2, and to conclude from Theorem 9 that the complexity of an ε-approximate path for Problem 12 for $t \in [0, \infty]$ is in $O\left(\frac{1}{\varepsilon}\right)$. Here the assumption that the distance problems associated to $K_c^{(1)}$ and $K_c^{(2)}$ are both separable holds trivially because $1/c > 0$.

In the case that we have more than two base kernels we can still apply the above approach if we fix the weights of all kernels except one. We can then navigate along the solution paths optimizing each kernel weight separately, and therefore try to find total weights with a hopefully best possible cross-validation accuracy. In Section 5.2 we report experimental results to determine the best combination of two kernels to achieve the highest prediction accuracy.

4.4 Minimum Enclosing Ball of Points under Linear Motion

Of interest from a more theoretical point of view is the following problem. Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in $\mathbb{R}^d$. The minimum enclosing ball (MEB) problem asks to find the smallest ball containing all points of P. The dual of the problem can be written [13] as

$$\begin{array}{ll} \max_x & x^T b - x^T A^T A x \\ \text{s.t.} & x \in S_n \end{array} \tag{13}$$

where $b = (b_i) = (p_i^T p_i)_{i \in [n]}$ and A is the matrix whose columns are the p_i.

Now we assume that the points each move with constant speed in a fixed direction, i.e., they move linearly as follows

$$p_i(t) = p_i + tv_i, \quad t \in [0, \infty)$$

where t can be referred to as time parameter. The MEB problem for moving points reads as

$$\begin{array}{ll} \max_x & x^T b(t) - x^T (P + tV)^T (P + tV) x \\ \text{s.t.} & x \in S_n \end{array} \tag{14}$$

where $b(t) = (b_i(t)) = ((p_i + tv_i)^T (p_i + tv_i))_{i \in [n]}$ and P is the matrix whose columns are the points p_i and V is the matrix whose columns are the vectors v_i. Problem 14 is a special case of the maximization version of Problem 6. Again, we are interested in the whole solution path, i.e. we want to track the center and the radius

$$r(t) = \sqrt{\hat{x}^T b(t) - \hat{x}^T (P + tV)^T (P + tV) \hat{x}} \qquad \text{with } \hat{x} \in S_n \text{ optimal}$$

of the MEB of the points $p_i(t)$ for $t \in [0, \infty)$ (or approximations of it). For an analysis of an approximate solution path we make use of the following observation.

Observation 1 *The interval $[0, \infty)$ can be subdivided into three parts: on the first sub-interval $r(t)$ is decreasing, on the second sub-interval, the radius $r(t)$ is constant, and on the third sub-interval, the radius is increasing.*

This can be seen as follows: consider the time when the radius of the MEB reaches its global minimum, just before the ball is expanding again. This is the point between the second and the third sub-interval. The points that cause the ball to expand at this point in time will prevent the ball from shrinking again in the future since the points move linearly. Thus the radius of the MEB will increase on the third sub-interval. By reversing the direction of time the same consideration leads to the observation that the radius of the MEB is decreasing on the first sub-interval.

We will consider each of the three sub-intervals individually. The second sub-interval can be treated like the standard MEB problem of non-moving points (MEB for the points v_i). Hence we only have to consider the first and the third sub-interval. We will only analyze the third sub-interval since the first sub-interval can be treated analogously with the direction of time reversed, i.e., the parameter t decreasing instead of increasing.

For the third sub-interval we know that the radius is increasing with time. We can shift the time parameter t such that we start with the third sub-interval at time $t = 0$. Let $r > 0$ be the radius $r(0)$ at time zero, i.e., we assume that the radius of the MEB never becomes zero. The case where the radius reaches 0 at some point is actually equivalent to the standard MEB problem for non-moving points. Without loss of generality we can scale all the vectors v_i such that the

MEB defined by the points v_i has radius r as well, because this just means scaling time. Without loss of generality we can also assume that the center of the MEB of the point sets P and V are both the origin. That is, $\|p_i\| \leq r$ and $\|v_i\| \leq r$. We have for $f_t(x) := x^T b(t) - x^T(P+tV)^T(P+tV)x$

$$\begin{aligned}(\nabla f_t(x))_i &= (p_i + tv_i)^T(p_i + tv_i) - 2(p_i + tv_i)^T(P+tV)x \\ &= \|p_i + tv_i - (P+tV)x\|^2 - x^T(P+tV)^T(P+tV)x\end{aligned}$$

and

$$\begin{aligned}x^T\nabla f_t(x) &= x^T\left(b(t) - 2(P+tV)^T(P+tV)x\right) \\ &= x^T b(t) - 2x^T(P+tV)^T(P+tV)x\end{aligned}$$

and hence

$$\begin{aligned}&(\nabla f_t(x))_i - x^T\nabla f_t(x) \\ &= \|p_i + tv_i - (P+tV)x\|^2 - \left(x^T b(t) - x^T(P+tV)^T(P+tV)x\right) \\ &= \|p_i + tv_i - (P+tV)x\|^2 - f_t(x)\end{aligned}$$

The partial derivative with respect to t of the above expression satisfies

$$\begin{aligned}&\frac{\partial}{\partial t}\left((\nabla f_t(x))_i - x^T\nabla f_t(x)\right) \\ &= \frac{\partial}{\partial t}\left(\|p_i + tv_i - (P+tV)x\|^2 - x^T b(t) + x^T(P+tV)^T(P+tV)x\right) \\ &= 2(p_i + tv_i - (P+tV)x)^T(v_i - Vx) - \frac{\partial}{\partial t}\sum x_i\|p_i + tv_i\|^2 + 2((P+tV)x)^T Vx \\ &= 2(p_i + tv_i - (P+tV)x)^T(v_i - Vx) - \sum 2x_i(p_i + tv_i)^T v_i + 2((P+tV)x)^T Vx\end{aligned}$$

Using the Cauchy-Schwarz inequality, $x \in S_n$, i.e., $\sum_{i=1}^n x_i = 1$, $x_i \geq 0$, and the fact that $\|p_i\|, \|v_i\| \leq r$ we get

$$\begin{aligned}\left|\frac{\partial}{\partial t}(\nabla f_t(x))_i - x^T\nabla f_t(x)\right| &\leq 2(r + tr + (r+tr))2r + 2(r+tr)r + 2(r+tr)r \\ &= 12r^2(1+t)\end{aligned}$$

Hence, by the mean value theorem we have that

$$(\nabla f_{t+\delta}(x) - \nabla f_t(x))_i - x^T(\nabla f_{t+\delta}(x)) - \nabla f_t(x)) \leq 12r^2(1+t+\delta)\delta.$$

From a similar calculation as above we obtain

$$|f_{t+\delta}(x) - f_t(x)| \leq 4r^2(1+t+\delta)\delta.$$

Now we can apply Lemma 3. Inequality 4 here simplifies to

$$12r^2(1+t+\delta)\delta + \varepsilon 4r^2(1+t+\delta)\delta \leq \varepsilon\left(1 - \frac{1}{\gamma}\right)r^2$$

since $f_t(x) \geq r^2$. Assuming $\varepsilon \leq 1$, we can set $\delta = \frac{\varepsilon}{32}\left(1 - \frac{1}{\gamma}\right)$ for $t, t+\delta \in [0,1]$. For the interval of $t, t+\delta \in [1,\infty)$ we apply the same trick as before and reduce it to the case of $t, t+\delta \in [0,1]$ by interchanging the roles of P and V. A short computation shows that an ε-approximation x at time $t \geq 1$ for the original optimization problem

$$\begin{array}{ll} \max_x & x^T b(t) - x^T (P+tV)^T (P+tV) x \\ \text{s.t.} & x \in S_n \end{array}$$

is an ε-approximation for the optimization problem

$$\begin{array}{ll} \max_x & x^T b'(t') - x^T (t'P+V)^T (t'P+V) x \\ \text{s.t.} & x \in S_n \end{array}$$

at time $t' = 1/t$, where $b'(t') = (b'_i(t')) = ((t'p_i + v_i)^T (t'p_i + v_i))_{i\in[n]}$, i.e., the roles of P and V have been interchanged. This is again due to the fact that the relative approximation guarantee is invariant under scaling. Summing the path complexities for all three sub-intervals that were considered in Observation 1, we conclude with the following theorem on the approximation path complexity for the minimum enclosing ball problem under linear motion:

Theorem 11 *The ε-approximation path complexity of the minimum enclosing ball Problem 14 for parameter $t \in [0,\infty)$ is at most*

$$2 \cdot 32 \frac{\gamma}{\gamma - 1} \frac{1}{\varepsilon} \quad + \quad 1 \quad + \quad 2 \cdot 32 \frac{\gamma}{\gamma - 1} \frac{1}{\varepsilon} \quad = \quad O\left(\frac{1}{\varepsilon}\right).$$

Since for the static MEB Problem 13, coresets of size $O(\frac{1}{\varepsilon})$ exist, see [4], we obtain the following corollary to Theorem 11.

Corollary 12 *There exists an ε-coreset of size $O(\frac{1}{\varepsilon^2})$ for Problem 14 that is globally valid under the linear motion, i.e., valid for all $t \geq 0$.*

The only other result known in this context is the existence of coresets of size $2^{O(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})}$ that remain valid under polynomial motions [2], and earlier, [1] have already proven the existence of coresets of size $O(1/\varepsilon^{2d})$ for the extent problem for moving points in $\mathbb{R}^d$, which includes the MEB problem as a special case.

5 Experimental Results

The parameterized framework from Section 3 is also useful in practice. For support vector machines and multiple kernel learning, we have implemented the approximation path Algorithms 1 and 2 in Java. As the internal blackbox optimization procedure in lines 1 and 7, we used the coreset variants [9] of the standard Gilbert's [10] and MDM [14] algorithms.

We have tested our implementations on the following standard binary classification datasets from the UCI repository[1]: ionosphere ($n = 280, d = 34$), breast-cancer ($n = 546, d = 10$), and MNIST 4k ($n = 4000, d = 780$). The timings were obtained by our single-threaded Java 6 implementation of MDM, using kernels but no caching of kernel evaluations, on an Intel C2D 2.4 GHz processor.

5.1 The Regularization Path of Support Vector Machines

Using the SVM formulation of Problem 10 (for the ℓ_2-SVM without offset), we compute approximate regularization paths for $c \in [c_{\min} = \frac{1}{100000}, c_{\max} = 100000]$ using the polynomial kernel $(\langle p_i, p_j \rangle + 1)^2$. As experimental results we report in Table 1 the following quantities: (a) the time T_{init} (in seconds) needed to compute an initial $\frac{\varepsilon}{\gamma}$-approximate solution as the starting point, (b) the time T_{path} (in seconds) needed to follow the entire ε-approximate regularization path, when starting from the initial solution, (c) for comparison the time T_3 (in seconds) needed to compute a static ε-approximate solution at the three fixed parameter values $c = c_{\min}, 1$ and $c_{\max}$, and (d) the path complexity, i.e. the number $\#int$ of obtained admissible parameter intervals of constant ε-approximate solutions along the path. The lower part of the table demonstrates the dependency of the path complexity on the choice of the parameter γ.

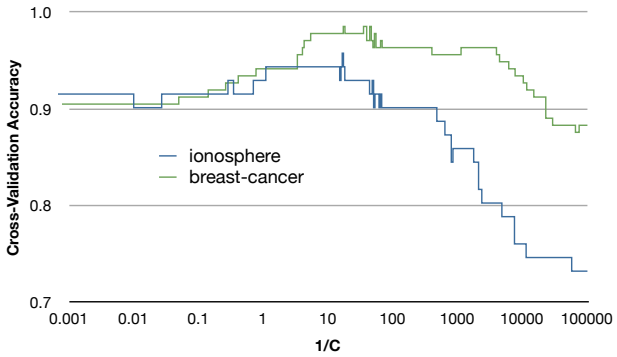

Figure 2: Continuous cross-validation along the ($\varepsilon = 0.2$)-approximate regularization path.

These experimental results show that the path complexity is indeed small if ε is not too small. We note that the method can be sped up further by using a more sophisticated internal optimization procedure. In practice, already relatively large values as for example $\varepsilon = 1$ are sufficient for good generalization

[1] All datasets are available from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`. In our experiments, all features were scaled to [-1,1]. For MNIST, the first 5000 'one' or 'seven' images of the original dataset were used.

$\gamma = 2$	*dataset*	'Forward' Algorithm 1			'Backward' Algorithm 2			
		#int	T_{path}	T_{init}	*#int*	T_{path}	T_{init}	T_3
$\varepsilon = 0.5$	ionosphere	53	5.2	2.8	98	4.3	0.3	3.3
	breast-cancer	65	4.7	3.3	102	5.2	0.4	3.3
	MNIST 4k	20	170.9	32.4	58	148.1	129.7	174.3
$\varepsilon = 0.1$	ionosphere	294	32.8	4.7	438	24.2	0.4	5.8
	breast-cancer	365	35.9	6.8	445	27.6	0.5	6.3
	MNIST 4k	103	837.1	52.3	274	722.7	169.7	264.0
$\varepsilon = 0.01$	ionosphere	3012	361.8	7.9	4251	251.7	0.6	9.9
	breast-cancer	3730	443.9	16.8	4307	305.9	0.7	16.5
	MNIST 4k	1030	8885.7	91.4	2692	7245.5	246.6	396.7

$\varepsilon = 0.1$	'Forward' Algorithm 1									
	$\gamma = 5$			$\gamma = 2$			$\gamma = 1.2$			
dataset	*#int*	T_{path}	T_{init}	*#int*	T_{path}	T_{init}	*#int*	T_{path}	T_{init}	T_3
ionosphere	188	53.6	5.9	294	32.8	4.7	808	26.0	4.2	5.8
breast-cancer	235	67.3	12.2	365	35.9	6.8	983	26.5	5.5	6.3
MNIST 4k	66	1487.9	70.9	103	837.1	52.3	288	656.9	47.5	264.0

Table 1: Path complexity and running times.

performance, as a primal-dual gap of ε implies that more than a $(1 - \frac{\varepsilon}{2})$-fraction of the best possible classification margin is already obtained [9].

From every dataset, a separate set of 1/5 of the original data points was kept for cross-validation. This means the computed classifier is evaluated on a small set of n_{cv} test points that have *not* been used solve the SVM optimization problem. Since our approximate regularization path has complexity at most $O\left(\frac{1}{\varepsilon}\right)$ (and we have a constant, ε-approximate solution on each admissible interval along the path), the cost of calculating all continuous cross-validation values, i.e., the percentages of correctly classified data points among the test points, along the entire regularization path is just $O\left(\frac{n_{cv}}{\varepsilon}\right)$ kernel evaluations. Cross-validation values along the path are shown in Figure 2.

5.2 Multiple Kernel Learning

In the multiple kernel learning setting of Problem 11, we used our implementation to compute approximate solution paths for $t \in [t_{\min} = \frac{1}{100000}, t_{\max} = 100000]$, for the problem to learn the best convex combination of the Gauss kernel $K^{(1)}$ with $\sigma = 8.5$, and the polynomial kernel $K^{(2)} = (\langle p_i, p_j \rangle + 1)^2$ on the same data sets as before. We chose a fixed regularization parameter value of $c = 1.5$. In Table 2 we report for Forward-Algorithm 1 (a) the time T_{init} (in seconds) needed compute an initial $\frac{\varepsilon}{\gamma}$-approximate solution as the starting point $t_{\min}$, (b) the time T_{path} (in seconds) needed to follow the entire ε-approximate regularization path, when starting from the initial solution, (c) for comparison

the time T_3 (in seconds) needed to compute a static ε-approximate solution at the three fixed parameter values $t = t_{\min}, 1$ and $t_{\max}$, and (d) the path complexity, i.e. the number $\#int$ of admissible parameter intervals with constant ε-approximate solutions along the path. Again a separate set of 1/5 of the original points was used to compute the resulting cross-validation values for an ε-approximate solution along the entire solution path, as shown in Figure 3.

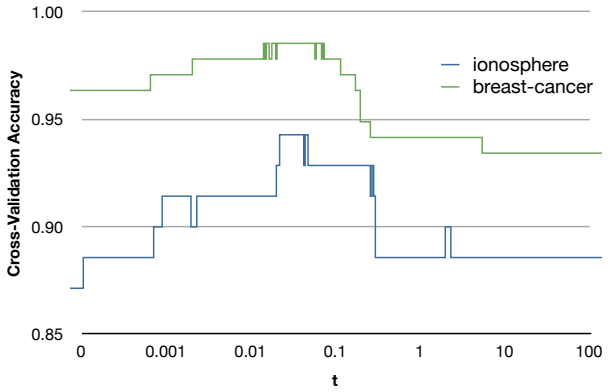

Figure 3: Continuous cross-validation along the ($\varepsilon = 0.2$)-approximate solution path.

$\gamma = 2$	*dataset*	$\#int$	T_{path}	T_{init}	T_3
$\varepsilon = 0.5$	ionosphere	53	14.1	4.4	6.8
	breast-cancer	71	4.0	2.4	3.7
	MNIST 4k	30	355.5	139.1	312.5
$\varepsilon = 0.1$	ionosphere	281	87.0	8.2	12.9
	breast-cancer	382	23.5	4.6	6.8
	MNIST 4k	150	2155.5	249.4	573.5

Table 2: Path complexity and running times

In practice, there are related methods that optimize a joint objective function over both the classifier weights and the combination of multiple kernels [3, 15]. These methods are experimentally fast, but are not directly comparable to ours as they do not obtain a solution path and are therefore unable to provide guarantees such as an optimal cross-validation value along a parameter path.

6 Conclusion

We have presented a framework to optimize convex functions over the unit simplex that are parameterized in one parameter. The framework is general,

simple and has been proven to be practical on a number of machine learning problems. Although it is simple it still provides improved theoretical bounds on known problems. In fact, we showed that our method is optimal up to a small constant factor.

References

[1] Pankaj Agarwal, Sariel Har-Peled, and Kasturi Varadarajan. Approximating extent measures of points. *Journal of the ACM*, 51(4):606–635, 2004.

[2] Pankaj Agarwal, Sariel Har-Peled, and Hai Yu. Embeddings of surfaces, curves, and moving points in euclidean space. *SCG '07: Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, 2007.

[3] Francis Bach, Gert Lanckriet, and Michael Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *ICML '04: Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[4] Mihai Bădoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Computational Geometry: Theory and Applications*, 40(1):14–22, 2007.

[5] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *SODA '08: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[6] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[7] Bernd Gärtner, Joachim Giesen, and Martin Jaggi. An exponential lower bound on the complexity of regularization paths. *arXiv*, cs.LG, 2009.

[8] Bernd Gärtner, Joachim Giesen, Martin Jaggi, and Torsten Welsch. A combinatorial algorithm to compute regularization paths. *arXiv*, cs.LG, 2009.

[9] Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. *SCG '09: Proceedings of the 25th Annual Symposium on Computational Geometry*, 2009.

[10] Elmer G Gilbert. An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM Journal on Control*, 4(1):61–80, 1966.

[11] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research*, 5:1391 – 1415, 2004.

[12] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, Mar 1952.

[13] Jirí Matousek and Bernd Gärtner. *Understanding and Using Linear Programming (Universitext)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[14] B. Mitchell, V. Dem'yanov, and V. Malozemov. Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, Jan 1974.

[15] A Rakotomamonjy, F Bach, and S Canu. Simplemkl. *Journal of Machine Learning Research*, 9, 2008.

[16] Saharon Rosset and Ji Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, 35(3):1012–1030, 2007.

[17] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

[18] Z Wu, A Zhang, C Li, and A Sudjianto. Trace solution paths for svms via parametric quadratic programming. *KDD '08 DMMT Workshop*, 2008.